

On the algorithmization of Janashia-Lagvilava matrix spectral factorization method

L.Ephremidze^{1,2}, F. Saied,¹ and I. Spitkovsky¹

¹ Division of Science and Mathematics, New York University Abu Dhabi (NYUAD), Saadiyat Island, P.O. Box 129188, Abu Dhabi, United Arab Emirates.

² A. Razmadze Mathematical Institute, I. Javakhishvili Tbilisi State University, 6, Tamarashvili st., Tbilisi 0177, Georgia. E-mail: le23@nyu.edu

Abstract. We consider three different ways of algorithmization of the Janashia-Lagvilava spectral factorization method. The first algorithm is faster than the second one, however, it is only suitable for matrices of low dimension. The second algorithm, on the other hand, can be applied to matrices of substantially larger dimension. The third algorithm is a superfast implementation of the method, but only works in the polynomial case under the additional restriction that the zeros of the determinant are not too close to the boundary. All three algorithms fully utilize the advantage of the method which carries out spectral factorization of leading principal submatrices step-by-step. The corresponding results of numerical simulations are reported in order to describe the characteristic features of each algorithm and compare them to other existing algorithms.

Keywords: Matrix spectral factorization, numerical algorithms.

Mathematics Subject Classification (2010): 65F30, 47A68.

1. INTRODUCTION

The Matrix Spectral Factorization (MSF) theorem [22], [10] asserts that if

$$(1) \quad S = \begin{pmatrix} s_{11}(t) & s_{12}(t) & \cdots & s_{1r}(t) \\ s_{21}(t) & s_{22}(t) & \cdots & s_{2r}(t) \\ \vdots & \vdots & \ddots & \vdots \\ s_{r1}(t) & s_{r2}(t) & \cdots & s_{rr}(t) \end{pmatrix},$$

$|t| = 1$, is a positive definite (a.e.) matrix function with integrable entries defined on the unit circle \mathbb{T} in the complex plane, $s_{ij}(t) \in L_1(\mathbb{T})$, and if the Paley-Wiener condition

$$(2) \quad \log \det S(t) \in L_1(\mathbb{T})$$

is satisfied, then (1) admits a spectral factorization

$$(3) \quad S(t) = S^+(t)(S^+(t))^*.$$

Here the entries of S^+ are square integrable functions, $s_{ij}^+ \in L_2(\mathbb{T})$, which can be extended analytically inside \mathbb{T} , i.e. s_{ij}^+ belongs to the Hardy space H_2 . Furthermore a spectral factor S^+ can be selected such that $\det S^+$ is an outer analytic function (see, e.g. [6]) and factorization (3) is unique (up to a constant right unitary multiplier)

under these conditions. S^+ is unique if we require $S^+(0)$ to be positive definite, and we always assume that it satisfies this condition as well.

In the scalar case, $r = 1$, the spectral factor $S^+ \in H_2$ can be explicitly written by the formula

$$(4) \quad S^+(z) = \exp \left(\frac{1}{4\pi} \int_0^{2\pi} \frac{e^{i\theta} + z}{e^{i\theta} - z} \log S(e^{i\theta}) d\theta \right).$$

If (1) is a Laurent polynomial matrix

$$(5) \quad S(t) = \sum_{k=-n}^n C_k t^k, \quad C_k \in \mathbb{C}^{r \times r},$$

then the spectral factor

$$(6) \quad S^+(t) = \sum_{k=0}^n A_k t^k, \quad A_k \in \mathbb{C}^{r \times r},$$

is a polynomial matrix of the same degree n (see e.g. [4] for an elementary proof).

Factorization (3) was first used in linear prediction theory of multidimensional stationary processes. Nowadays, it is widely known that MSF plays a crucial role in the solution of various applied problems for multiple-input and multiple-output systems in Communications and Control Engineering [14]. Recently MSF became an important step in non-parametric estimations of Granger causality used in Neuroscience [2], [21]. These applications require the matrix coefficients of analytic S^+ to be determined, at least approximately, for a given matrix function S . Therefore, starting with Wiener's original efforts [23] to create a sound computational method of MSF, dozens of different algorithms have appeared in the literature (see the survey papers [16], [18] and the references therein, and also [1], [11] for more recent results).

A novel approach to the solution of the MSF problem, without imposing any additional restriction on S besides the necessary and sufficient condition (2) for the existence of spectral factorization, was originally developed by Janashia and Lagvilava in [12] for 2×2 matrices. This approach was subsequently extended to matrices of arbitrary dimension in [13]¹. Results of preliminary numerical simulations based on the proposed method were presented in the same paper [13]. However, a closer look at possible algorithmization ways of this method revealed further advantages. In fact, numerical simulations carried out by the improved algorithms produced much better results than it was reported in [13]. That this development required additional investigations is not surprising, as all methods of MSF are quite demanding and, as it is mentioned in [16]: “the numerical properties of each method strongly depend on the way it is algorithmized”.

In the present paper, after a general description of the Janashia-Lagvilava method (Sections III and IV), we describe three different algorithms of MSF based on this method: JLE-1 (Section VI), JLE-2 (Section VII), and JLE-3 (Section VIII). As it was mentioned above, the method is general and also suitable for non-rational matrices. However, since in practical applications the data is finite, we concentrate our attention on the polynomial case. Furthermore, JLE-algorithm 3 is designed only for polynomial matrices (5) with the additional restriction that $\det S(t) \neq 0$ for $t \in \mathbb{T}$ (the so-called non-singular case). Its theoretical justification is not yet completed. Nevertheless, due

¹This method obtained USPTO patent recently: No. 9,318,232; issued April 19, 2016.

to its superfast speed, we present JLE-3 in the current form. The JLE-algorithm 1 is faster than JLE-2 and it can deal with singular case as well, but it is only suitable for low dimensional matrices. JLE-algorithm 2 can be applied for much larger matrices, depending on available time and accuracy. In Section IX, we demonstrate the ability of the method to factorize singular matrices. In Section X, we compare with Wilson's MSF method. The results of provided numerical simulations are presented in Section XI and concluding remarks are given in Section XII. We emphasize that the proposed MSF method uses the existing scalar spectral factorization algorithms, whenever they are called for, and does not attempt to improve upon these.

2. NOTATION

Let $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$ be the open unit disk, and $\mathbb{T} = \partial\mathbb{D}$ be the unit circle. As usual, $L_p = L_p(\mathbb{T})$, $0 < p < \infty$, denotes the Lebesgue space of p -integrable complex functions defined on \mathbb{T} (L_∞ is the space of essentially bounded functions). For $p \geq 1$, $\|f\|_p$ is the usual norm. $H_p = H_p(\mathbb{D})$, $0 < p \leq \infty$, is the Hardy space of analytic functions in \mathbb{D} ,

$$H_p := \left\{ f \in \mathcal{A}(\mathbb{D}) : \sup_{r < 1} \int_0^{2\pi} |f(re^{i\theta})|^p d\theta < \infty \right\}$$

(H_∞ is the space of bounded analytic functions), and $L_p^+ = L_p^+(\mathbb{T})$ denotes the class of their boundary functions. A function $f \in H_p$ is called outer, denoted $f \in H_p^O$, if

$$f(z) = c \cdot \exp \left(\frac{1}{2\pi} \int_0^{2\pi} \frac{e^{i\theta} + z}{e^{i\theta} - z} \log |f(e^{i\theta})| d\theta \right), \quad |c| = 1.$$

The n th Fourier coefficient of an integrable function $f \in L_1(\mathbb{T})$ is denoted by $c_k\{f\}$. For $p \geq 1$, $L_p^+(\mathbb{T})$ coincides with the class of functions from $L_p(\mathbb{T})$ whose Fourier coefficients with negative indices are equal to zero.

The set of trigonometric polynomials is denoted by \mathcal{P} , i.e. $f \in \mathcal{P}$ if f has only a finite number of nonzero Fourier coefficients. In particular, for integers $m \leq n$, let $\mathcal{P}_{\{m,n\}} := \{f \in \mathcal{P} : c_k\{f\} = 0 \text{ whenever } k < m \text{ or } k > n\}$ and, for a non-negative integer N , let $\mathcal{P}_N^+ := \mathcal{P}_{\{0,N\}}$, $\mathcal{P}_N^- := \mathcal{P}_{\{-N,0\}}$. Obviously, $f \in \mathcal{P}_N^+ \Leftrightarrow \bar{f} \in \mathcal{P}_N^-$. For a function $f \in L_1$ with Fourier expansion $f \sim \sum_{n \in \mathbb{Z}} c_k t^k$ (or for a formal Fourier series) and positive integer N , let \mathbb{P}_N^+ , \mathbb{P}_N^- , and \mathbb{Q}_N^+ be the following projection operators:

$$\mathbb{P}_N^+[f] = \sum_{k=0}^N c_k t^k, \mathbb{P}_N^-[f] = \sum_{k=0}^N c_{-k} t^{-k}, \text{ and } \mathbb{Q}_N^-[f] = \sum_{k=1}^N c_{-k} t^{-k}$$

If M is a matrix, then \overline{M} denotes the matrix with complex conjugate entries and $M^* := \overline{M}^T$. Furthermore, $\mathbb{C}^{m \times m}$, $L_p(\mathbb{T})^{m \times m}$, etc., denote the set of $m \times m$ matrices with the entries from \mathbb{C} , $L_p(\mathbb{T})$, etc. If $S \in \mathbb{C}^{r \times r}$ is a matrix (function) and $m \leq r$, then $S_{[m]}$ stands for the upper-left $m \times m$ submatrix of S ($S_{[0]}$ is assumed to be 1) and $S_{[1:r,m]}$ stands for m th column of S . Matrices like $S_{[1:r-1,m]}$ or $S_{[1:r-1,1:m]}$ are defined accordingly. The matrix $S_{[i,j]}$ is obtained from S by deleting the i th row and j th column.

For a polynomial $p(t) = \sum_{k=0}^n p_k t^k$, let $\|p\| = \sup_{0 \leq k \leq n} |p_k|$, and for a polynomial matrix $P = (P_{ij})_{i,j=1}^r$, let $\|P\| = \sup_{1 \leq i,j \leq r} \|P_{ij}\|$.

A matrix $M \in \mathbb{C}^{r \times r}$ is called positive definite if $X^* M X > 0$ for all $0 \neq X \in \mathbb{C}^{r \times 1}$, and $S \in L_1(\mathbb{T})^{r \times r}$ is called positive definite if it is positive definite for a.a. $t \in \mathbb{T}$.

A matrix function $U \in L^\infty(\mathbb{T})^{r \times r}$ is called unitary if

$$(7) \quad U(t)U^*(t) = I_r \text{ a.e.},$$

where I_r stands for $r \times r$ identity matrix.

$\mathbf{0}_{r \times m}$ and $\mathbf{1}_{r \times m}$ stand for $r \times m$ matrices with all entries equal to 0 and 1, respectively. Using Matlab's notation, if $A \in \mathbb{C}^{r \times m_1}$ and $B \in \mathbb{C}^{r \times m_2}$, then $[A \ B]$ is $r \times (m_1 + m_2)$ matrix, while if $A \in \mathbb{C}^{r_1 \times m}$ and $B \in \mathbb{C}^{r_2 \times m}$, then $[A; B] = [A^T \ B^T]^T$ is $(r_1 + r_2) \times m$ matrix.

For a column vector $\mathbf{a} = [a_0 \ a_1 \ \dots \ a_l]^T \in \mathbb{C}^{(l+1) \times 1}$ and a positive integer $m \in \mathbb{N}$, let $T(\mathbf{a}; m)$ be the $(l + m + 1) \times (m + 1)$ Toeplitz matrix with the first column $[\mathbf{a}; \mathbf{0}_{m \times 1}] \in \mathbb{C}^{l+m+1}$ and the first row $[a_0 \ \mathbf{0}_{1 \times m}] \in \mathbb{C}^{1 \times (m+1)}$.

We say that a sequence of matrix functions S_n , $n = 1, 2, \dots$ is convergent to a matrix function S (in some sense) if the entries of S_n are convergent to the corresponding entries of S (in this sense).

Finally, δ_{ij} stands for the Kronecker delta, i.e. $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

3. GENERAL DESCRIPTION OF THE METHOD

The first step of the MSF method proposed in [13] is the triangular factorization of
(1)

$$(8) \quad S(t) = M(t)M^*(t),$$

where $M(t)$ is the lower triangular matrix

$$(9) \quad M(t) = \begin{pmatrix} f_1^+(t) & 0 & \dots & 0 & 0 \\ \xi_{21}(t) & f_2^+(t) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{r-1,1}(t) & \xi_{r-1,2}(t) & \dots & f_{r-1}^+(t) & 0 \\ \xi_{r1}(t) & \xi_{r2}(t) & \dots & \xi_{r,r-1}(t) & f_r^+(t) \end{pmatrix},$$

$\xi_{ij} \in L_2(\mathbb{T})$, $f_i^+ \in H_2^O$. The spectral factor S^+ is represented in the form

$$(10) \quad S^+(t) = M(t)\mathbf{U}_2(t)\mathbf{U}_3(t)\dots\mathbf{U}_r(t) \cdot U.$$

Here each \mathbf{U}_m is a block matrix function

$$(11) \quad \mathbf{U}_m(t) = \begin{pmatrix} U_m(t) & \mathbf{0}_{m \times (r-m)} \\ \mathbf{0}_{(r-m) \times m} & I_{r-m} \end{pmatrix},$$

where $U_m(t)$ is a special unitary matrix function of the form

$$(12) \quad U_m(t) = \begin{pmatrix} u_{11}(t) & u_{12}(t) & \dots & u_{1,m-1}(t) & u_{1m}(t) \\ u_{21}(t) & u_{22}(t) & \dots & u_{2,m-1}(t) & u_{2m}(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{m-1,1}(t) & u_{m-1,2}(t) & \dots & u_{m-1,m-1}(t) & u_{m-1,m}(t) \\ \overline{u_{m1}(t)} & \overline{u_{m2}(t)} & \dots & \overline{u_{m,m-1}(t)} & \overline{u_{mm}(t)} \end{pmatrix},$$

with

$$(13) \quad u_{ij} \in L_+^\infty, \quad \text{and} \quad \det U(t) = 1 \text{ a.e.}$$

(for reasons explained in [7] such matrices can as well be called “wavelet matrices”). Furthermore, for each $m = 2, 3, \dots, r$,

$$(14) \quad S_{[m]}^+ = (M\mathbf{U}_2\mathbf{U}_3 \dots \mathbf{U}_m)_{[m]}$$

is a spectral factor of $S_{[m]}$. In particular, $S_0^+ := M\mathbf{U}_2\mathbf{U}_3 \dots \mathbf{U}_r$ is a spectral factor of (1), and the constant unitary matrix U in (10) makes S^+ positive definite in the origin, namely (see [5, formula (54)])

$$(15) \quad U = (S_0^+(0))^{-1} \sqrt{S_0^+(0)(S_0^+(0))^*}.$$

To obtain unitary matrix function (12) for each $m = 2, 3, \dots, r$ recurrently, we consider a matrix function

$$(16) \quad F_m(t) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ \zeta_1(t) & \zeta_2(t) & \zeta_3(t) & \dots & \zeta_{m-1}(t) & f_m^+(t) \end{pmatrix},$$

where the last row of (16) consists of the first m entries of the m th row of the product

$$(17) \quad M_{m-1} := M\mathbf{U}_2\mathbf{U}_3 \dots \mathbf{U}_{m-1},$$

and then obtain a matrix function (12), (13) such that (see [5, Lemma 4])

$$(18) \quad F_m U_m \in L_2^+(\mathbb{T})^{m \times m}.$$

Particularly, we have

$$(19) \quad (M_{m-1})_{[m]} = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & S_{[m-1]}^+(t) & & \vdots \\ & & & 0 \\ \zeta_1(t) & \dots & \zeta_{m-1}(t) & f_m^+(t) \end{bmatrix} = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & S_{[m-1]}^+(t) & & \vdots \\ & & & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} F_m(t)$$

and

$$(20) \quad S_{[m]}^+(t) = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & S_{[m-1]}^+(t) & & \vdots \\ & & & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} F_m(t) U_m(t).$$

In order to achieve (18), one needs to consider the following system of conditions (see [13, formula (15)])

$$(21) \quad \begin{cases} \zeta_1(t)x_m^+(t) - f_m^+(t)\overline{x_1^+(t)} \in L_2^+, \\ \zeta_2(t)x_m^+(t) - f_m^+(t)\overline{x_2^+(t)} \in L_2^+, \\ \vdots \\ \zeta_{m-1}(t)x_m^+(t) - f_m^+(t)\overline{x_{m-1}^+(t)} \in L_2^+, \\ \zeta_1(t)x_1^+(t) + \zeta_2(t)x_2^+(t) + \dots + \zeta_{m-1}(t)x_{m-1}^+(t) + f_m^+(t)\overline{x_m^+(t)} \in L_2^+, \end{cases}$$

and columns of (12) are m independent solutions of (21).

To construct (12) approximately the following procedures should be performed:

For a large positive N , let $F_m^{\{N\}}$ be the matrix function (16) with the last row replaced by

$$(\zeta_1^{\{N\}}, \zeta_2^{\{N\}}, \dots, \zeta_{m-1}^{\{N\}}, f_m^+),$$

where

$$\zeta_j^{\{N\}}(t) := \sum_{k=-N}^{\infty} c_k \{\zeta_j\} t^k, \quad j = 1, 2, \dots, m-1.$$

Then one can find the unitary matrix function $U_m^{\{N\}}$ of the form (12) such that $\det U_m^{\{N\}}(t) = 1$, $U_m^{\{N\}}(1) = I_m$, $u_{ij} \in \mathcal{P}_N^+$ and $F_m^{\{N\}} U_m^{\{N\}} \in \mathcal{P}_N^+$ (see [13, Theorem 1]). In particular, the columns of $U_m^{\{N\}}$ are m independent solutions of the system (21) where $\zeta_1, \zeta_2, \dots, \zeta_{m-1}$ are replaced by $\zeta_1^{\{N\}}, \zeta_2^{\{N\}}, \dots, \zeta_{m-1}^{\{N\}}$, and they can be actually found by solving a single system of $(N+1) \times (N+1)$ linear algebraic equations with m different right-hand sides (see the proof of Theorem 1 in [13]). Details of the computation are given in Section IV.

One can prove that $U_m^{\{N\}} \rightarrow U_m$ at least in measure as $N \rightarrow \infty$, which guarantees that (see [5, Theorem 2])

$$M U_2 U_3 \dots U_m^{\{N\}} \rightarrow M U_2 U_3 \dots U_m \quad \text{in } L_2.$$

4. CONSTRUCTION OF WAVELET MATRICES

In this section we provide the details of computation of the unitary matrix function $U_N := U_m^{\{N\}}$ for a given matrix function (16). N and m are assumed fixed throughout this section.

Let

$$\mathbb{P}_N^+[f_m^+](t) = \sum_{k=0}^N d_k t^k, \quad \mathbb{Q}_N^+[\zeta_i](t) = \sum_{k=1}^N \gamma_{in} t^{-k}, \quad \text{and} \quad \mathbb{P}_N^+[1/f_m^+](t) = \sum_{k=0}^N b_k t^k.$$

(Note that the knowledge of $\mathbb{P}_N^+[f_m^+]$ is sufficient to determine $\mathbb{P}_N^+[1/f_m^+]$.) Suppose D^{-1} is the upper triangular Toeplitz matrix with the first row

$$(22) \quad (b_0, b_1, \dots, b_N),$$

and Γ_i , $i = 1, 2, \dots, m-1$ is the upper triangular Hankel matrix with the first row

$$(23) \quad (0, \gamma_{i,1}, \gamma_{i,2}, \dots, \gamma_{iN})$$

(see [13, (26)]) and let

$$(24) \quad \Theta_i = D^{-1} \Gamma_i, \quad i = 1, 2, \dots, m-1.$$

Note that Θ_i is the upper triangular Hankel matrix (see [13, (33)]) with the first row

$$(25) \quad \Lambda_i := (\eta_{i0}, \eta_{i1}, \dots, \eta_{iN}),$$

where $\sum_{k=0}^N \eta_{in} t^{-k} = \mathbb{P}_N^- \left[\sum_{k=0}^N b_k t^k \cdot \sum_{k=1}^N \gamma_{in} t^{-k} \right]$.

Take

$$(26) \quad \Delta = \sum_{i=1}^{m-1} \Theta_i \Theta_i^* + I_{N+1},$$

which is a positive definite matrix (with all eigenvalues ≥ 1), and solve the same system of equations (see (25))

$$(27) \quad \Delta X = \Lambda_i^T$$

with m different right hand sides corresponding to $i = 1, 2, \dots, m$. Here it is assumed that $\Lambda_m = (1, 0, 0, \dots, 0)$. The matrix (26) has a displacement structure of rank m , namely

$$\Delta - Z\Delta Z^* = \sum_{i=1}^{m-1} \Lambda_i \Lambda_i^* + \mathcal{E} \mathcal{E}^*$$

has rank m , where Z is the upper triangular $(N+1) \times (N+1)$ matrix with 1's on the first up-diagonal and 0's elsewhere (i.e. a Jordan block with eigenvalue 0) and $\mathcal{E} = (0, 0, \dots, 0, 1)^T \in \mathbb{C}^{N+1,1}$ (see [13, Appendix]). Therefore its triangular factorization $\Delta = LDL^*$ can be achieved in $O(mN^2)$ operations instead of $O(N^3)$ as explained e.g. in [14, Appendix F] without even constructing the matrix Δ (just using the $(N+1) \times m$ matrix $[\Lambda_1, \Lambda_2, \dots, \Lambda_{m-1}, \mathcal{E}]$).

Let the solution of (27) be $X_i = (a_{i0}, a_{i1}, \dots, a_{iN})^T$, and denote

$$v_{mi}(t) := \sum_{k=0}^N a_{in} t^k, \quad i = 1, 2, \dots, m,$$

Suppose also

$$v_{ij}(t) = \mathbb{P}_N^+ \left[\sum_{k=0}^N \bar{\eta}_n t^k \cdot \sum_{k=0}^N \overline{a_{in}} t^{-k} \right] - \delta_{ij},$$

$1 \leq i \leq m-1$, $1 \leq j \leq m$, and let

$$V(t) = \begin{pmatrix} v_{11}(t) & v_{12}(t) & \cdots & v_{1,m-1}(t) & v_{1m}(t) \\ v_{21}(t) & v_{22}(t) & \cdots & v_{2,m-1}(t) & v_{2m}(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{m-1,1}(t) & v_{m-1,2}(t) & \cdots & v_{m-1,m-1}(t) & v_{m-1,m}(t) \\ \overline{v_{m1}(t)} & \overline{v_{m2}(t)} & \cdots & \overline{v_{m,m-1}(t)} & \overline{v_{mm}(t)} \end{pmatrix}.$$

Then (see [13, (51)])

$$U_N(t) = V(t) \cdot V^{-1}(0).$$

It is proved in [13] that $V(0)$ is nonsingular and the condition number of this matrix is estimated in [8].

5. A SHORTCUT IN THE RECURSIVE STEP

As it was mentioned in Section III, in order to perform m th recursive step in the proposed MSF method, we need only to consider

$$S_{[m-1]}^+ = (M_{m-1})_{[m-1]}$$

(see (17) and (14)), which has already been constructed (at least approximately) and the first m entries in the m th row of M_{m-1}

$$(28) \quad (M_{m-1})_{[m,1:m]} = (\zeta_1, \zeta_2, \dots, \zeta_{m-1}, f_m^+)$$

(see (16)). Because of the block structure of matrices in (11), the entry f_m^+ is the same as in (9). Thus it can be computed by the formula (see [13, formula (56)])

$$(29) \quad f_m^+ = \frac{(\det S_{[m]})^+}{(\det S_{[m-1]})^+}$$

((\cdot)⁺ stands for the scalar spectral factorization (4)).

Since $S = M_{m-1} M_{m-1}^*$ (see (8), (17), and (7)) and particularly

$$(30) \quad (S)_{[m]} = (M_{m-1})_{[m]} (M_{m-1})_{[m]}^*$$

(see (19)), we have

$$(31) \quad S_{[m-1]}^+ \cdot (\zeta_1, \zeta_2, \dots, \zeta_{m-1})^* = S_{[1:m-1, m]}.$$

Therefore, instead of computing matrices M_m for each $m = 2, 3, \dots, r-1$ by (17), we can directly compute the entries $\zeta_1, \zeta_2, \dots, \zeta_{m-1}$ from (31).

Having computed the functions $\zeta_1, \zeta_2, \dots, \zeta_{m-1}$, one can find $|f_m^+|^2$ from the formula (see (30))

$$(32) \quad \sum_{j=1}^{m-1} |\zeta_j|^2 + |f_m^+|^2 = s_{mm}.$$

Therefore, an alternative way of computing (29) is the scalar spectral factorization of $s_{mm} - \sum_{j=1}^{m-1} |\zeta_j|^2$.

In the next three sections we present three different implementations of the described algorithm for polynomial data (5), followed by the results of corresponding numerical simulations.

6. JLE-ALGORITHM 1

This algorithm relies on computation of polynomial matrix determinant. Namely, for a polynomial matrix of order n

$$(33) \quad P(t) = \sum_{k=0}^n B_k t^k, \quad B_k \in \mathbb{C}^{m \times m},$$

$\det P$ is a polynomial of order mn . Therefore, having evaluated $\det P(t)$ at $mn+1$ DFT nodes $t_l = \exp\left(\frac{2\pi i l}{mn+1}\right)$, $l = 0, 1, \dots, mn$, the coefficients of $\det P$ can be computed by interpolation, namely computing the inverse DFT of $[\det P(t_0), \dots, \det P(t_{mn})]$.

This algorithm of polynomial matrix determinant computation is fast and accurate for matrices of small dimension. However, the algorithm suffers from severe round-off errors and the accuracy is destroyed for large dimensional matrices. For example, with a standard double precision in Matlab, we have found a computation error in the formula

$$\|\det(P_1 P_2) - \det P_1 \det P_2\|$$

as small as 10^{-8} for randomly selected polynomial matrices P_1 and P_2 of degree $n = 10$ and dimension $m = 10$, and as large as 10^9 for ones with $n = 20$ and $m = 15$. The reason of such increase is that the coefficients of $\det P$ become very large (at least for randomly selected coefficients B_k in (33)) and floating point machine arithmetic loses significant digits. Therefore JLE-algorithm 1 (with input (5) and output (6)) is

suitable for small dimensional matrices ($r < 20$ and $n < 25$). Its basic computational procedures are described below.

Procedure 1. Compute the diagonal entries of the triangular factor (9) by the formula (29), where $m = 1, 2, \dots, r$. Each f_m^+ can be represented as a rational function p_m/q_m , where $p_m \in \mathcal{P}_{mn}^+$ and $q_m \in \mathcal{P}_{(m-1)n}^+$. In addition, the denominator is free of zeros inside \mathbb{T} , and f_m^+ is free of poles on \mathbb{T} (since $f_m^+ \in L_2^+(\mathbb{T})$).

For the scalar spectral factorization of $\det S_{[m]}$, we first apply exp-log implementation by using FFT [9] and then we improve the accuracy by using 4-5 iterations of Wilson's scalar factorization algorithm [24].

Procedure 2. For $m = 2, 3, \dots, r$, assume that $S_{[m-1]}^+$ has already been (approximately) constructed as an $(m-1) \times (m-1)$ polynomial matrix of degree n and perform the following steps.

Step 1. Compute ζ_j , $j = 1, 2, \dots, m-1$, by the Cramer's rule from equation (31). In particular, each ζ_j will be of the form p/q , where $p \in \mathcal{P}_{\{-n, (m-1)n\}}$ and $q \in \mathcal{P}_{(m-1)n}^+$, again with q free of zeros inside \mathbb{T} and ζ_j free of poles on \mathbb{T} . Note that ζ_j -s can be computed in parallel.

Step 2. Select a large positive integer N . Theoretically, as $N \rightarrow \infty$, the computed spectral factor $\hat{S}_{[m]}^+$ converges to exact $S_{[m]}^+$ (assuming that all previous factors including $S_{[m-1]}^+$ are computed exactly). However, in practise we never achieve an exact result. Nevertheless, the accuracy

$$(34) \quad \|S_{[m]} - \hat{S}_{[m]}^+(\hat{S}_{[m]}^+)^*\|$$

can be controlled and the value of N can be increased, if necessary, at each intermediate stage, in order to achieve a satisfactory approximation in the final result.

Step 3. From obtained representations of ζ_j , $j = 1, 2, \dots, m-1$, and f_m^+ as rational functions, find

$$\zeta_j^{\{N\}} := \mathbb{Q}_N^+[\zeta_j] + \mathbb{P}_n^+[\zeta_j] = \sum_{k=-N}^n c_k \{\zeta_j\} t^k$$

and

$$f_m^{\{N\}} := \mathbb{P}_{N+n}^+[f_m^+] = \sum_{k=0}^{N+n} c_k \{f_m^+\} t^k.$$

We do this by the standard division algorithm of two polynomials, utilizing the advantages of denominator being free from zeros inside \mathbb{T} and function having no poles on \mathbb{T} .

Step 4. Using $(\zeta_1^{\{N\}}, \zeta_2^{\{N\}}, \dots, \zeta_{m-1}^{\{N\}}, f_m^{\{N\}})$ as the last row of (16), construct a unitary matrix function $U_N := U_m^{\{N\}}$ as it is described in Section IV.

Step 5. Consider the product

$$S_{[m]}^+ \approx \begin{pmatrix} & & & 0 \\ & & & \vdots \\ & S_{[m-1]}^+ & & 0 \\ \zeta_1^{\{N\}} & \zeta_2^{\{N\}} & \dots & \zeta_{m-1}^{\{N\}} & f_m^{\{N\}} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ u_{m-1,1} & u_{m-1,2} & \dots & u_{m-1,m} \\ \overline{u_{m1}} & \overline{u_{m2}} & \dots & \overline{u_{mm}} \end{pmatrix}$$

(the last matrix is $U_m^{\{N\}}$), where all coefficients of polynomials in the right-hand side product with indices outside the range $[0, n]$ are neglected (since we know that the

exact $S_{[m]}^+$ is matrix polynomial of degree n). Therefore, $S_{[m-1]}^+$ can be separately multiplied by the first $m-1$ rows of $U_m^{\{N\}}$ and then its last row can be multiplied by $U_m^{\{N\}}$.

Procedure 3. For $m = r$, $S_{[r]}^+$ is an approximate spectral factor of S . We can multiply $S_{[r]}^+$ by the constant unitary matrix U defined by (15) (taking $S_{[r]}^+$ instead of S_0^+) to obtain S^+ .

7. JLE-ALGORITHM 2

In this implementation, computations of polynomial matrix determinants are avoided. Consequently much higher dimensional matrices can be factorized accurately by this algorithm at the expense of large computer memory usage.

Procedure 1. Compute a scalar spectral factor f_1^+ of s_{11} by using the same exp-log and Wilson's methods as in Procedure 1 of JLE-algorithm 1.

Procedure 2. For $m = 2, 3, \dots, r$, assume that $S_{[m-1]}^+$ has already been (approximately) constructed as an $(m-1) \times (m-1)$ polynomial matrix of degree n and perform the following steps.

Step 1. Take a large number of DFT nodes, usually 2^κ , where $10 \leq \kappa \leq 23$: $t_l = \exp\left(\frac{2\pi i l}{2^\kappa}\right)$, $l = 0, 1, \dots, 2^\kappa - 1$. This κ becomes another tuning parameter in the algorithm (along with N), which can be selected and changed during recursive steps in order to improve the accuracy (34).

Step 2. For each node t_l , $l = 0, 1, \dots, 2^\kappa - 1$, evaluate the matrices $S_{[m-1]}^+(t_l)$ and $S_{[1:m-1, m]}(t_l)$, and solve the following system of linear equations (see (31)):

$$(35) \quad S_{[m-1]}^+(t_l) \cdot X = S_{[1:m-1, m]}(t_l).$$

We have $(\zeta_1(t_l), \zeta_2(t_l), \dots, \zeta_{m-1}(t_l)) = X_l^*$, where X_l is the solution of (35).

If it happens that the system (35) is singular or ill conditioned, then we can apply the continuity of functions ζ_j and assume that $X_l = X_{l-1}$.

When standard routines are well optimized (as it is in Matlab), this step is not as time-consuming as it might appear at the first glance.

Step 3. Compute $|f_m^+(t_l)|^2$, $l = 0, 1, \dots, 2^\kappa - 1$, from the formula (32)

Step 4. Select a large positive integer N , and using the values of $|f_m^+|^2$ at DFT nodes, perform an approximate scalar spectral factorization to reconstruct

$$f_m^{\{N\}} := \sum_{k=0}^{N+n} c_k \{f_m^+\} t^k.$$

For this step, one can use the exp-log method of scalar spectral factorization which utilizes the boundary values of a spectral density.

The integer N has a natural bound $2^\kappa - n$ in this situation, however an optimal ratio (from $1/10$ to $1/50$) of $N/2^\kappa$ should be selected in order to achieve a good accuracy.

Step 5. From the values of ζ_j at DFT nodes t_l , $l = 0, 1, \dots, 2^\kappa - 1$, reconstruct (approximately)

$$(36) \quad \zeta_j^{\{N\}} := \sum_{k=-N}^n c_k \{\zeta_j\} t^k$$

by using the inverse FFT and selecting corresponding coefficients.

The remaining steps are the same as Steps 4 and 5 in JLE-algorithm 1, including Procedure 3.

8. JLE-ALGORITHM 3

This implementation utilizes formulas (20), (29), and (31) for $m = r$:

$$(37) \quad S^+(t) = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & S_{[r-1]}^+(t) & & \vdots \\ & & & 0 \\ \zeta_1(t) & \zeta_2(t) & \dots & \zeta_{r-1}(t) & f_r^+(t) \end{bmatrix} \begin{bmatrix} u_{11}(t) & u_{12}(t) & \dots & u_{1r}(t) \\ u_{21}(t) & u_{22}(t) & \dots & u_{2r}(t) \\ \vdots & \vdots & \vdots & \vdots \\ u_{r-1,1}(t) & u_{r-1,2}(t) & \dots & u_{r-1,r}(t) \\ \overline{u_{r1}(t)} & \overline{u_{r2}(t)} & \dots & \overline{u_{rr}(t)} \end{bmatrix},$$

$$(38) \quad f_r^+(t) = \det S^+(t) / \det S_{[r-1]}^+(t),$$

and

$$(39) \quad [\zeta_1(t), \zeta_2(t), \dots, \zeta_{r-1}(t)] \cdot (S_{[r-1]}^+(t))^* = S_{[r,1:r-1]}(t).$$

Let $U(t) = U_r(t)$ be the last matrix in (37). Then, for $j \leq r$, it follows from (37) that

$$(40) \quad S_{[r-1]}^+(t) \cdot U_{[1:r-1,j]}(t) = S_{[1:r-1,j]}^+(t),$$

and furthermore

$$(41) \quad S_{[r-1]}^+(t) \cdot U_{[r,j]}(t) = S_{[r,j]}^+(t).$$

Since $U(t)$ is a unitary matrix ($U^{-1}(t) = U^*(t)$) and $\det U(t) = 1$, it follows that $u_{r,j}(t) = \det U_{[r,j]}(t)$ and, taking into account (41), we get

$$(42) \quad u_{r,j}(t) = \det S_{[r,j]}^+(t) / \det S_{[r-1]}^+(t).$$

It also follows from (37) that

$$(43) \quad [\zeta_1(t), \zeta_2(t), \dots, \zeta_{r-1}(t)] \cdot U_{[1:r-1,j]}(t) + f_r^+(t) \overline{u_{r,j}(t)} = S_{rj}^+(t).$$

Substituting into (43) $[\zeta_1, \dots, \zeta_{r-1}] = S_{[r,1:r-1]} \cdot (S_{[r-1]}^+)^{-*}$ (see (39)), $U_{[1:r-1,j]} = (S_{[r-1]}^+)^{-1} \cdot S_{[1:r-1,j]}^+$ (see (40)), (38), and (42), and taking into account that $S_{[r-1]} = S_{[r-1]}^+ (S_{[r-1]}^+)^*$, we get

$$S_{[r,1:r-1]}(t) \cdot (S_{[r-1]}(t))^{-1} \cdot S_{[1:r-1,j]}^+(t) + \frac{\det S^+(t) \det S_{[r,j]}^+(t)}{\det S_{[r-1]}(t)} = S_{rj}^+(t).$$

Consequently,

$$(44) \quad S_{[r,1:r-1]}(t) \cdot \text{Cof} \{S_{[r-1]}(t)\}^T \cdot S_{[1:r-1,j]}^+(t) + (\det S(t))^+ \cdot \overline{\det S_{[r,j]}^+(t)} = S_{rj}^+(t) \det S_{[r-1]}(t),$$

where it is assumed that $(\det S(t))^+$ can be found from $\det S(t)$, as the problem is reduced to the scalar spectral factorization.

In the equation (44), $S_{[r,1:r-1]}$, $\text{Cof} \{S_{[r-1]}\}^T$, $(\det S(t))^+$ and $\det S_{[r-1]}$ are assumed to be the known (matrix) functions, and $S_{[1:r-1,j]}^+$, $\det S_{[r,j]}^+$ and S_{rj}^+ are unknown (matrix) functions.

Assume now that S is a matrix polynomial of degree n (see (5)), i.e. $S \in (\mathcal{P}_{\{-n,n\}})^{r \times r}$. Let us observe that for functions in (44) we have:

$$S_{[r,1:r-1]} \in (\mathcal{P}_{\{-n,n\}})^{1 \times (r-1)}; \text{Cof} \{S_{[r-1]}\}^T \in (\mathcal{P}_{\{-n(r-2),n(r-2)\}})^{(r-1) \times (r-1)};$$

$$S_{[1:r-1,j]}^+ \in (\mathcal{P}_{\{0,n\}})^{(r-1) \times 1}; (\det S)^+ \in \mathcal{P}_{\{0, rn\}}; \overline{\det S_{[r,j]}^+} \in \mathcal{P}_{\{-(r-1)n, 0\}}; S_{rj}^+ \in \mathcal{P}_{\{0,n\}},$$

and $\det S_{[r-1]} \in \mathcal{P}_{\{-(r-1)n, (r-1)n\}}$. Thus all products in (44) have the range of indices of (nonzero) Fourier coefficients in $[-(r-1)n, rn]$. If we equate the corresponding coefficients in these products, we get $2rn - n + 1$ linear algebraic equations with respect to coefficients of unknown (matrix) polynomials $S_{[1:r-1,j]}^+$, $\det S_{[r,j]}^+$ and S_{rj}^+ . The total number of these coefficients is $(r-1)(n+1) + \{(r-1)n+1\} + (n+1) = 2rn - n + r + 1$.

We can factorize $S(t)$ at a single point on the unit circle, say $t = 1$, and getting the representation $S(1) = S^+(1)(S^+(1))^*$, we can assume that $[S_{[1:r-1,j]}^+(1) S_{rj}^+(1)]^T$ is the j -th column of $S^+(1)$. This gives the additional r conditions on coefficients of (matrix) polynomials $S_{[1:r-1,j]}^+$ and S_{rj}^+ , and thus additional r equations. In the end we get the same number of linear equations and unknowns $2rn - n + r + 1$.

The basic computational procedures of the algorithm are described below.

Step 1. Compute the polynomial determinants $\det S(t)$ and $\det S_{[r-1]}(t)$ by the method described in JLE-1.

Step 2. Compute the scalar spectral factor $(\det S(t))^+$ by the method described in Procedure 1 of JLE-1.

Step 3. Compute $\text{Cof} \{S_{[r-1]}(t)\}^T$ by evaluating it at $N = 2n(r-1) + 1$ DFT nodes $t_l = \exp(\frac{2\pi i l}{N})$, $l = 0, 1, 2, \dots, N-1$, by the formula $\text{Cof} \{S_{[r-1]}(t_l)\}^T = \det S_{[r-1]}(t_l) (S_{[r-1]}(t_l))^{-1}$ and then use the inverse Fourier transform.

Step 4. Multiply matrix polynomials $S_{[r,1:r-1]}$ and $\text{Cof} \{S_{[r-1]}\}^T$.

Let $(\det S(t))^+ = \sum_{k=0}^{rn} a_k t^k$, $t^{(r-1)n} \det S_{[r-1]}(t) = \sum_{k=0}^{2(r-1)n} b_k t^k$, and

$$t^{(r-1)n} S_{[r,1:r-1]}(t) \text{Cof} \{S_{[r-1]}\}^T(t) = \sum_{k=0}^{2(r-1)n} C_k t^k = \left[\sum_{k=0}^{2(r-1)n} c_k^{\{1\}} t^k \dots \sum_{k=0}^{2(r-1)n} c_k^{\{r-1\}} t^k \right],$$

$C_k \in \mathbb{C}^{1 \times (r-1)}$, $c_k^{\{j\}} \in \mathbb{C}$. Introduce also the notation: $\mathbf{a} = [a_0 \ a_1 \ \dots \ a_{rn}]^T \in \mathbb{C}^{(2rn+1) \times 1}$; $\mathbf{b} = [b_0 \ b_1 \ \dots \ b_{2(r-1)n}]^T \in \mathbb{C}^{(2(r-1)n+1) \times 1}$; $\mathbf{c}^{\{j\}} = [c_0^{\{j\}} \ c_1^{\{j\}} \ \dots \ c_{2(r-1)n}^{\{j\}}]^T \in \mathbb{C}^{(2(r-1)n+1) \times 1}$, $j = 1, 2, \dots, r-1$.

Step 5. Construct the $(2rn - n + 1) \times (2rn - n + r + 1)$ matrix $\Delta_0 = [\Delta_1 \ \Delta_2 \ \Delta_3]$, where $\Delta_1 = [T(\mathbf{c}^{\{1\}}; n) \ T(\mathbf{c}^{\{2\}}; n) \ \dots \ T(\mathbf{c}^{\{r-1\}}; n)] \in \mathbb{C}^{(2rn-n+1) \times (r-1)(n+1)}$, $\Delta_2 = -T(\mathbf{b}; n) \in \mathbb{C}^{(2rn-n+1) \times (n+1)}$, and $\Delta_3 = T(\mathbf{a}; (r-1)n) \in \mathbb{C}^{(2rn-n+1) \times ((r-1)n+1)}$ and then the $(2rn - n + r + 1) \times (2rn - n + r + 1)$ matrix $\Delta = [\Delta_1 \ \Delta_2 \ \Delta_3; \mathbf{I} \ \mathbf{0}_{r \times ((r-1)n+1)}]$, where $\mathbf{I} \in \mathbb{C}^{r \times r(n+1)}$ is the $r \times r$ block identity matrix with entries $\mathbf{1}_{1 \times (n+1)}$ on the block diagonal and $\mathbf{0}_{1 \times (n+1)}$ elsewhere.

Step 6. Perform the Cholesky factorization of the positive definite matrix $S(1) = S^+(1)(S^+(1))^*$ and assume that $S^+(1) = [h_1 \ h_2 \ \dots \ h_r]$, where $h_j \in \mathbb{C}^{r \times 1}$.

Step 7. For each $j = 1, 2, \dots, r$, solve the $(2rn - n + r + 1) \times (2rn - n + r + 1)$ system of equations

$$(45) \quad \Delta X = \Lambda_j,$$

with right-hand sides $\Lambda_j = [\mathbf{0}_{(2rn-n+1) \times 1} ; h_j]$, and denote the respective solution by $X_j = [x_0^{\{j\}} x_1^{\{j\}} \dots x_{2rn-n+r}^{\{j\}}]^T$.

Step 8. Set a spectral factor $S_0^+ = (s_{ij}^+)_{i,j=1}^r$, where $s_{ij}^+(t) = \sum_{k=0}^n x_{(n+1)(i-1)+k}^{\{j\}} t^k$

Step 9. Find S^+ by $S_0^+ U$, where U is defined by the formula (15).

Since we know the existence of decomposition (37), the solution to equation (45) exists for each j . However it might happen that $\det \Delta = 0$. Furthermore, computer simulations suggest that Δ is nonsingular whenever $\det S(t) \neq 0$ for each $t \in \mathbb{T}$ and Δ is singular whenever $\det S(t) = 0$ for some $t \in \mathbb{T}$. Therefore JLE-3 works under the additional condition $\det S(t) > 0$ for $t \in \mathbb{T}$. If this condition holds, but zeros of $\det S$ are rather close to the boundary, the matrix Δ might become ill-conditioned. In such situations, the solutions of (45) are inaccurate and approximation to S^+ is lost. The techniques of solution of ill-conditioned systems might be useful, however we have not investigated this question yet. As numerical simulations show in Section IX, JLE-algorithm 3 can satisfactory factorize random matrices with $r = 6$ and $n = 20$, which might be useful in certain applications to Mobile Communications [19].

9. FACTORIZATION OF SINGULAR MATRICES

Symmetric positive matrix polynomials which are chosen randomly or obtained by channel estimation in wireless communication are usually non-singular, i.e. their determinants do not vanish on \mathbb{T} . However, in certain optimal control and wavelet design problems, one encounters a need to factorize singular matrices. It is well known that all MSF methods have difficulties in this situation and some of them cannot handle zeros on the unit circle at all. Obviously, convergence of JLE algorithms also slows down in singular cases. However, if we fully utilize the ability of Janashia-Lagvilava's method to decompose a large scale problem into smaller parts and deal with any arising difficulties by intermediate interventions, in number of cases we can substantially improve the performance of the algorithm. In this section we demonstrate this advantage by factorizing specific singular matrices.

First, consider a test matrix from [13] whose spectral factorization is known beforehand:

$$(46) \quad \begin{pmatrix} 2z^{-1} + 6 + 2z & 11z^{-1} + 22 + 7z \\ 7z^{-1} + 22 + 11z & 38z^{-1} + 84 + 38z \end{pmatrix} = \begin{pmatrix} 2 + z & 1 \\ 7 + 5z & 3 + z \end{pmatrix} \begin{pmatrix} 2 + z^{-1} & 7 + 5z^{-1} \\ 1 & 3 + z^{-1} \end{pmatrix}$$

This matrix is very simple, but its determinant, $-z^{-2} + 2 - z^2$, has two double zeros on the boundary.

When data was fed into "standard" JLE-algorithm 1 with 5 iterations in scalar spectral factorization of $\det S$ by Wilson's algorithm (see Sect. 6, Procedure 1), we get 4 correct digits. When we increase the number of the iterations up to 45, the maximum optimum value, we get 7 correct digits. If we compute the determinant by the direct formula $\det S = s_{11}s_{22} - s_{12}s_{21}$, avoiding the minimal round-off errors introduced with computation of the determinant by FFT (see Section 6), then we get 14 correct digits. All these computations take less than 0.01 seconds as the matrix is very small and and

it suffices to select the parameter N as small as 20. We observed that Wilson's MSF algorithm (see the next section) can perform factorization (46) with no more than 6 correct digits (with optimum parameter $\kappa = 19$) which takes around 3 minutes.

Next we factorize a small size 2×2 matrix

$$(47) \quad S(z) = \sum_{k=-3}^3 C_k z^k = \begin{pmatrix} s_{11}(z) & s_{12}(z) \\ s_{21}(z) & s_{22}(z) \end{pmatrix},$$

where $s_{11}(z) = -\frac{1-4\bar{\alpha}}{64}z^{-3} + \frac{1+4\alpha}{64}z^{-1} + 1 + \frac{1+4\alpha}{64}z - \frac{1-4\bar{\alpha}}{64}z^3$; $s_{12}(z) = \frac{\bar{\alpha}}{16}z^{-3} - \frac{\alpha}{16}z^{-1} + \frac{\alpha}{16}z - \frac{\bar{\alpha}}{16}z^3$; $s_{21}(z) = s_{12}(1/z)$; and $s_{22}(z) = \frac{1-4\bar{\alpha}}{64}z^{-3} - \frac{1+4\alpha}{64}z^{-1} + 1 - \frac{1+4\alpha}{64}z + \frac{1-4\bar{\alpha}}{64}z^3$; with $\alpha = 4 + \sqrt{15}$ and $\bar{\alpha} = 4 - \sqrt{15}$. This matrix is singular and, furthermore, its determinant has an explicit form $\det S(z) = \frac{8\bar{\alpha}-1}{4096}(z+1)^4(z-1)^4(z+i)^2(z-i)^2$. Its spectral factorization $S(z) = \sum_{k=0}^3 A_k z^k \sum_{k=0}^3 A_k^T z^{-k}$ is required for construction of the so called SA4 multiwavelet [20] which possess certain nice properties. The realization of these properties depends on the accuracy by which the coefficients A_k are computed. The efforts to factorize (47) with a maximal possible accuracy by the Youla-Kazanjan method [27] is described in [15], where the error $err_1 = \|S(z) - \sum_{k=0}^3 \hat{A}_k z^k \sum_{k=0}^3 \hat{A}_k^T z^{-k}\| = 4.086 \cdot 10^{-8}$ is achieved. (As the exact values of A_k are unknown in this situation, this error is used to estimate the accuracy $\|A_k - \hat{A}_k\|$.) As we checked, this performance cannot be improved by the Wilson MSF method either. In fact, the error cannot be reduced to lower than 10^{-5} by the method (with optimal tuning parameter $\kappa = 18$: see Section 10).

When we ran JLE-1 with the matrix S and increase the number of iterations in the scalar factorization step up to 60 (see Procedure 1), we obtain the error $err_2 = 4.373 \cdot 10^{-5}$. However, if we cancel out the common roots in the triangular factorization (8) and factorize the determinant $\det S$ manually we achieve the error $err_3 = 1.843 \cdot 10^{-14}$. In these computations, it is sufficient to take the tuning parameter $N = 100$ and so the consumed time is very small (less than 0.1 seconds).

In general, when a singular polynomial (with a zero on \mathbb{T}) is factorized in the scalar case, the best way to deal with the singularity is to factor out the zeros with unit modulus. This procedure is more demanding in the matrix case (see [17, p. 67]). The above examples demonstrate that Janashia-Lagvilava method is capable of reducing a problem of the singularity of a spectral matrix density to the level of scalar factorization. In fact, the method has already been used to improve the coefficients of other well-known multiwavelets as well by effective factorization of related singular matrices which will be the topic of another paper.

10. COMPARISON WITH WILSON'S ALGORITHM

Wilson's method of MSF appeared in the 70's of the last century [25], [26]. Since then, several authors claimed that they obtained MSF algorithms with reduced computational complexity (see [16, p. 1077], [14, p. 206]). These are algorithms based on the solution of algebraic Riccati equations and some of them are implemented in Matlab. As a consequence, in our attempts to compare Janashia-Lagvilava algorithm with other existing methods of MSF, we did not originally consider the Wilson method and only concentrated our attention on those methods which were implemented in Matlab (see [13, Sect. VI]). However, recently we learned that Prof. Rangarajan and his collaborators, who apply MSF in Neuroscience [2], [3], developed an efficient implementation of Wilson's method which works rather fast.

This implementation takes data matrix in frequency domain. Nevertheless, this idea can be easily translated for matrices given in time domain. In particular, for a matrix (5) with given coefficients C_k , $k = 0, 1, \dots, N$, we select κ as a tuning parameter and find 2^κ values of the matrix function S in DFT nodes: $S(t_0), \dots, S(t_{2^\kappa})$, where $t_j = \exp\left(\frac{2\pi i j}{2^\kappa}\right)$. Then we use the Wilson's recurrent formula

$$(48) \quad S_{k+1}^+ = S_k^+ [(S_k^+)^{-1} S(S_k^+)^{-*} + I]^+$$

with initial data $S_0 = \sqrt{C_0}$. After performing sufficient iterations, we return back to the time domain and approximately compute the coefficients A_k of (6). Here, like other minor improvements we introduced in the implementation of Wilson's method, we empirically observed that the upper triangular constant matrix S_τ in formula (3.2) in [26] can be omitted in (48). Such implementation of Wilson's algorithm essentially works as efficient as JLE-1 and frequently better than JLE-2. In addition, a flexible combination of Janashia-Lagvilava and Wilson methods can be sometimes useful.

11. NUMERICAL SIMULATIONS

The computer code for implementation of JLE-algorithms was written in Matlab in order to test them numerically. A laptop with characteristics Intel(R) Core(TM) i7-4600U CPU (2 cores, 4 threads), 2.40GHz, RAM 8.00Gb was used and some of the tests were performed on the HPC cluster "Dalma" at NYUAD.

For all numerical simulations of MSF algorithms randomly selected polynomial matrices have been used. Namely, for given matrix dimension r and polynomial degree n , a random polynomial matrix $\sum_{k=0}^n A_k t^k$, $A_k \in [-1, 1]^{r \times r}$, has been chosen, and positive definite (on \mathbb{T}) matrix polynomial $S(t) = \sum_{k=0}^n A_k t^k \sum_{k=0}^n A_k^* t^{-k}$ has been approximately factorized. In rare occasions, which are emphasized below, some deterministic efforts have been introduced in order to artificially improve the properties of S . The error

$$(49) \quad err = \|S - \hat{S}^+(\hat{S}^+)^*\|$$

is used to estimate the accuracy of the factorization since there is no other way to decide how close is \hat{S}^+ to S^+ .

The basic problem in order to demonstrate the most effective performance of the constructed algorithms was an empirical selection of tuning parameters (N for JLE-1, N and κ for JLE-2, and κ and the number of iterations for Wilson's algorithm) which would make an optimal trade-off between the available memory, the computation time and the accuracy.

For realistic applications, automatic selection of the optimal tuning parameters during the factorization remains a challenging problem.

When different algorithms are compared, it is assumed that they were run with the same data.

We start with JLE-3 which has the advantage that it contains no tuning parameters. Below we demonstrate its performance within the range of polynomial matrices for which it is applicable. The tuning parameters in JLE-1 and Wilson have been selected so as to achieve the same accuracy as in JLE-3. Beyond the indicated range of matrix dimension m and polynomial degree n the accuracy (49) of JLE-3 becomes unsatisfactory. (In all tables below, $r \times n$ indicates that a $r \times r$ test matrix was selected

with Laurent polynomial entries of degree n having nonzero coefficients indexed from $-n$ to n).

Table I
Performance of JLE-3

	matrix size	time <i>sec</i>	accu- racy	matr. size	time <i>sec</i>	accu- racy
JLE-3	4×30	0.052	10^{-8}	6×20	0.051	10^{-6}
JLE-1	—	0.315	10^{-8}	—	0.576	10^{-6}
Wilson	—	1.108	10^{-8}	—	0.694	10^{-6}
JLE-3	8×10	0.051	10^{-6}	10×5	0.044	10^{-6}
JLE-1	—	0.359	10^{-6}	—	0.352	10^{-6}
Wilson	—	0.419	10^{-6}	—	0.360	10^{-6}

Next we compare JLE-1 and Wilson within the range of matrices where JLE-1 operates well. The tuning parameter $N = 5mn$ has been taken for m th recursion in JLE-1 and κ has been selected in Wilson so as to achieve the same accuracy as in JLE-1.

Table II
Comparison of JLE-1 and Wilson

	matrix size	tuning parameters	time <i>sec</i>	accuracy
JLE-1	10×100	$N = 500\ m$	6.35	$1.63 \cdot 10^{-7}$
Wilson	—	$\kappa = 12$; Iter = 23	7.91	$1.42 \cdot 10^{-7}$
JLE-1	15×20	$N = 100\ m$	2.67	$6.12 \cdot 10^{-8}$
Wilson	—	$\kappa = 11$; Iter = 25	3.95	$1.76 \cdot 10^{-8}$

Next we factorize random 100×100 matrices (with polynomial degree $n = 30$) by JLE-2 and Wilson. We tried to factorize such matrices with accuracy that is acceptable in practice, namely $error = 10^{-4}$, and selected the tuning parameters accordingly. A substantial drop in the accuracy has been observed at the final step of recursion $m = 100$ in JLE-2 and it was observed that Wilson can factorize the 99×99 leading submatrix of S much more easily than S itself. We empirically explain this phenomenon by the following reason: the probability for zeros of $\det S_{[m]}$ to be very close to \mathbb{T} (in which case all spectral factorization algorithms become slowly convergent) is higher for $m = r$ than for $m < r$ (however no theoretical proofs has been attempted). Therefore, in a variant of our implementation, we have combined JLE-2 by Wilson which resulted in certain improvements.

Table III
Comparison of JLE-2 and Wilson

100×30	tuning parameters	time	accuracy
JLE-2	$N = 2400e^{0.15(m-100)} + 80e^{0.01m}$	94.1	$5 \cdot 10^{-4}$
Wilson	$\kappa = 11$; Iter = 14	85.3	$1 \cdot 10^{-4}$
Wil+JLE	99×99 by Wilson+JLE-2	43.3	$2 \cdot 10^{-4}$

When we added artificially I_r to a random matrix S in order to avoid zeros close to \mathbb{T} , we achieved the same accuracy within improved computation time. We display the results below.

Table IV
Comparison of JLE-2 and Wilson

$S_{rand} + I$	tuning parameters	time	accuracy
JLE-2	$N \approx 100e^{0.02m}$	24.1	$4 \cdot 10^{-4}$
Wilson	$\kappa = 8$; Iter = 9	6.57	$3 \cdot 10^{-4}$
Wil+JLE	99×99 by Wilson+JLE-2	6.23	$5 \cdot 10^{-4}$

In the end we demonstrate that “good” matrices of dimension as large as 700×700 can be factorized with accuracy $error = 10^{-3}$ which is acceptable in practice and within the available computer memory (120GB of one node at “Dalma” in our situation). With respect to time usage, the advantage of Wilson’s MSF method is evident in this case. The reason is that JLE-2 requires the tuning parameter N to be selected very large at the last recursive steps in order to achieve the given accuracy. However, JLE-2 algorithm still can be invoked to analyze and overcome the problem when Wilson’s method is unable to factorize a matrix obtained from real applications.

Table V
Factorization of large matrices

700×30 $S_{rnd} + I$	tuning parameters	time (hours)	accu- racy	RAM
JLE-2	$N \approx 200e^{0.003m}$	2 : 50	$4 \cdot 10^{-3}$	80GB
Wilson	$\kappa = 11$; Iter = 17	1 : 08	$3 \cdot 10^{-3}$	120GB

12. CONCLUSIONS

Matrix spectral factorization is widely used in modern control theory and wireless communications. Furthermore, improved algorithms of MSF may lead to new areas to which they could be successfully applied. In the present paper, we consider three different algorithms based on Janashia-Lagvilava method, which may be competitive with other existing MSF algorithms. A general description of their computational capabilities, as well as a comparison to Wilson’s MSF algorithm, are provided by means of numerical simulations.

13. ACKNOWLEDGMENTS

The authors are thankful for an opportunity to run part of the tests using the High Performance Computing resources at New York University Abu Dhabi.

REFERENCES

- [1] A. Böttcher and M. Halwass, *A Newton method for canonical Wiener-Hopf and spectral factorization of matrix polynomials*, Electron. J. Linear Algebra **26** (2013), 873–897. MR 3192406
- [2] M. Dhamala, G. Rangarajan, and M Ding, *Analyzing information flow in brain networks with nonparametric granger causality*, NeuroImage **41** (2008), 354–362.
- [3] ———, *Estimating granger causality from fourier and wavelet transforms of time series data*, Physical Review Letters **100** (2008), 018701.
- [4] L. Ephremidze, *An elementary proof of the polynomial matrix spectral factorization theorem*, Proc. Roy. Soc. Edinburgh Sect. A **144** (2014), no. 4, 747–751. MR 3233753
- [5] L. Ephremidze, G. Janashia, and E. Lagvilava, *On approximate spectral factorization of matrix functions*, J. Fourier Anal. Appl. **17** (2011), no. 5, 976–990. MR 2838115 (2012h:47039)
- [6] L. Ephremidze and E. Lagvilava, *Remark on outer analytic matrix-functions*, Proc. A. Razmadze Math. Inst. **152** (2010), 29–32. MR 2663529
- [7] ———, *On compact wavelet matrices of rank m and of order and degree N* , J. Fourier Anal. Appl. **20** (2014), no. 2, 401–420. MR 3200928

- [8] L. Ephremidze, N. Salia, and I. Spitkovsky, *Some aspects of a novel matrix spectral factorization algorithm*, Proc. A. Razmadze Math. Inst. **166** (2014), 49–60. MR 3300615
- [9] T. N. T. Goodman, Ch. A. Micchelli, G. Rodriguez, and S. Seatzu, *Spectral factorization of Laurent polynomials*, Adv. Comput. Math. **7** (1997), no. 4, 429–454. MR 1470294
- [10] H. Helson and D. Lowdenslager, *Prediction theory and Fourier series in several variables*, Acta Math. **99** (1958), 165–202. MR 0097688 (20 #4155)
- [11] A. Jafarian and J. G. McWhirter, *A novel method for multichannel spectral factorization*, Proc. Europ. Signal Process. Conf. (2012), 27–31.
- [12] G. Janashia and E. Lagvilava, *A method of approximate factorization of positive definite matrix functions*, Studia Math. **137** (1999), no. 1, 93–100. MR 1735630 (2000m:15015)
- [13] G. Janashia, E. Lagvilava, and L. Ephremidze, *A new method of matrix spectral factorization*, IEEE Trans. Inform. Theory **57** (2011), no. 4, 2318–2326. MR 2809092 (2012d:65077)
- [14] T. Kailath, B. Hassibi, and A. H. Sayed, *Linear estimation*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1999, Prentice-Hall Information and System Sciences Series.
- [15] V. Kolev, T. V. Cooklev, and F. Keinert, *Matrix spectral factorization - SA4 multiwavelet*, Preprint.
- [16] V. Kučera, *Factorization of rational spectral matrices: A survey of methods*, in Proc. IEEE Int. Conf. Control, Edinburgh **2** (1991), 1074–1078.
- [17] G. S. Litvinchuk and I. M. Spitkovskii, *Factorization of measurable matrix functions*, Operator Theory: Advances and Applications, vol. 25, Birkhäuser Verlag, Basel, 1987, Translated from the Russian by Bernd Luderer, With a foreword by Bernd Silbermann. MR 1015716
- [18] A. H. Sayed and T. Kailath, *A survey of spectral factorization methods*, Numer. Linear Algebra Appl. **8** (2001), no. 6-7, 467–496, Numerical linear algebra techniques for control and signal processing. MR 1848590 (2002j:93039)
- [19] A. B. Sergienko and V. P. Klimentyev, *Scma detection with channel estimation error and resource block diversity*, in Proc. Int. Siberian Conf. Control and Communications, (SIBCON) (2016), DOI: 10.1109/SIBCON.2016.7491765.
- [20] J. Y. Tham, L. Shen, S. L. Lee, and H. H. Tan, *A general approach for analysis and application of discrete multiwavelet transforms*, IEEE Trans. Signal Process. **48** (2000), no. 2, 457–464. MR 1746064
- [21] X. Wen, G. Rangarajan, and M. Ding, *Multivariate granger causality: an estimation framework based on factorization of the spectral density matrix*, Phil. Trans. R. Soc. A **371**: 20110610. (2013).
- [22] N. Wiener and P. Masani, *The prediction theory of multivariate stochastic processes. I. The regularity condition*, Acta Math. **98** (1957), 111–150. MR 0097856 (20 #4323)
- [23] ———, *The prediction theory of multivariate stochastic processes. II. The linear predictor*, Acta Math. **99** (1958), 93–137. MR 0097859 (20 #4325)
- [24] G. Wilson, *Factorization of the covariance generating function of a pure moving average process*, SIAM J. Numer. Anal. **6** (1969), 1–7. MR 0253561
- [25] G. Tunnicliffe Wilson, *The factorization of matricial spectral densities*, SIAM J. Appl. Math. **23** (1972), 420–426. MR 0331843
- [26] ———, *A convergence theorem for spectral factorization*, J. Multivariate Anal. **8** (1978), no. 2, 222–232. MR 497596
- [27] D. C. Youla and N. N. Kazanjian, *Bauer-type factorization of positive matrices and the theory of matrix polynomials orthogonal on the unit circle*, IEEE Trans. Circuits and Systems **CAS-25** (1978), no. 2, 57–69. MR 0469461